



Fusion de réseaux de contraintes qualitatives par morceaux

Jean-François Condotta, Souhila Kaci, Pierre Marquis, Nicolas Schwind

► To cite this version:

Jean-François Condotta, Souhila Kaci, Pierre Marquis, Nicolas Schwind. Fusion de réseaux de contraintes qualitatives par morceaux. Cinquièmes Journées Francophones de Programmation par Contraintes, Jun 2009, Orléans, France. pp.135-145. hal-00387820

HAL Id: hal-00387820

<https://hal.science/hal-00387820>

Submitted on 25 May 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fusion de réseaux de contraintes qualitatives par morceaux

Jean-François Condotta, Souhila Kaci, Pierre Marquis et Nicolas Schwind

Université Lille-Nord de France, Artois, F-62307 Lens

CRIL, F-62307 Lens

CNRS UMR 8188, F-62307 Lens

{condotta,kaci,marquis,schwind}@cril.fr

Résumé

La représentation et le raisonnement sur le temps et l'espace est une problématique importante pour de nombreuses applications de l'intelligence artificielle : la compréhension du langage naturel, la navigation de robots, etc. Ces trente dernières années de nombreux formalismes dits qualitatifs ont été proposés dans le but de représenter un ensemble d'entités spatiales ou temporelles et leur relations. Nous nous intéressons dans cet article au problème de la fusion de réseaux de contraintes qualitatives (RCQ en abrégé). Nous décrivons un algorithme de fusion qui construit un RCQ représentant une vision globale de l'ensemble des RCQ, éventuellement conflictuels, donnés en entrée. Cet algorithme est générique car il est indépendant du formalisme qualitatif dans lequel sont définis les RCQ. En comparaison avec une précédente approche pour la fusion de RCQ, la méthode que nous proposons est plus efficace puisqu'elle est basée sur une fusion locale des contraintes portant sur les mêmes variables (sans considérer dans un premier temps leurs interactions avec les autres contraintes). Nous définissons plusieurs opérateurs de fusion de contraintes de telle sorte que le RCQ construit résultant possède certaines propriétés d'un point de vue logique. Nous montrons en outre comment rendre le processus de fusion traitable en imposant certaines restrictions.

Abstract

Representing space and time and reasoning about them are important issues for many AI applications. In the past three decades numerous qualitative formalisms have been proposed for this purpose. In this paper the problem of merging qualitative constraints networks (QCNs) is addressed. We point out a merging algorithm which computes a QCN representing a global view of the input set of (possibly conflicting) QCNs. This algorithm is generic in the sense that it does not depend on a specific qualitative formalism. Compared to previous

approaches to QCNs merging, the efficiency of our method comes from the fact that it first merges locally the constraints of the input QCNs bearing on the same pairs of variables (without considering in this first step their interactions with the other constraints). We define several constraint merging operators in a way to ensure that the induced QCNs merging operator satisfies some expected properties from a logical standpoint. We also point out some restrictions which make the corresponding QCNs merging operators tractable.

1 Introduction

La représentation et le raisonnement sur des entités spatiales ou temporelles représente une tâche importante dans de nombreux domaines de l'Intelligence Artificielle. [2, 8, 7]. Ces 30 dernières années de nombreux formalismes qualitatifs ont été développés pour représenter un ensemble d'entités spatiales ou temporelles et leur relations. Dans le domaine spatial, les relations utilisées peuvent être de type topologiques [15] (lorsque les entités considérées représentent des régions de points) ou basées sur une relation d'ordre [12]. Dans le domaine temporel, les relations d'Allen [1] permettent de représenter différentes relations de précédence entre des intervalles de la droite des rationnels. Plus récemment des formalismes basés sur les relations d'Allen ont été proposés [4, 3], ainsi que des combinaisons de formalismes qualitatifs [6].

La plupart de ces formalismes utilisent des réseaux de contraintes qualitatives (RCQ) pour représenter un ensemble d'entités spatiales ou temporelles et leurs positions relatives.

Dans certaines applications, en particulier de type multi-agents, l'information spatiale ou temporelle provient de différentes sources, c'est-à-dire que chaque

source fournit un RCQ représentant ses connaissances sur l'ensemble des configurations d'un même ensemble d'entités. En raison de la multiplicité des sources, les RCQ fournis sont généralement conflictuels. Il faut alors mettre en œuvre un processus de fusion pour résoudre les conflits. Par exemple, considérons l'exemple suivant. Trois cours communs (Bases de données, Prolog et Algorithmique) sont prévus pour un groupe d'étudiants. Ces derniers ont la possibilité d'exprimer leur préférences sur la planification de ces cours. Supposons qu'un premier étudiant veuille suivre Bases de données avant Prolog, et Prolog avant Algorithmique, et qu'un autre étudiant préfère acquérir des notions d'Algorithmique avant de suivre le cours Bases de données. Clairement aucun planning ne peut satisfaire pleinement tous les étudiants, et nous devons trouver un compromis cohérent pour résoudre les conflits.

Ces quinze dernières années de nombreux opérateurs de fusion ont été définis et étudiés dans le cadre de la logique propositionnelle [16, 9]. Une adaptation directe de certains de ces travaux pour la fusion de RCQ a été proposée dans [5]. Elle consiste à définir un opérateur de fusion qui prend en entrée un ensemble fini de RCQ représentant l'information fournie par les différentes sources, et retourne comme résultat un ensemble cohérent d'informations spatiales ou temporelles représentant une vision globale de ces RCQ. En revanche il n'existe pas d'implémentation pratique de cette approche en raison sa complexité importante.

Dans cet article nous présentons une méthode efficace pour la fusion de RCQ. Cette nouvelle méthode consiste à fusionner localement les contraintes des RCQ portant sur les mêmes paires de variables, en exploitant les opérateurs de fusion propositionnelle basés sur des calculs de distance [9]. Nous définissons une procédure de fusion de RCQ de manière à obtenir comme résultat de la fusion un RCQ *cohérent*.

La suite de cet article est organisée comme suit : le paragraphe suivant présente les définitions et notations utilisées à propos des formalismes spatio-temporels, et en particulier à propos des RCQ. Dans le paragraphe 3 nous introduisons au travers d'un exemple le problème de la fusion d'un ensemble de RCQ conflictuels. Nous décrivons dans le paragraphe 4 notre nouvelle méthode de fusion de RCQ, exploitant un opérateur de fusion local sur les contraintes basé sur un calcul de distances. Dans le paragraphe 5, nous montrons comment améliorer l'algorithme de fusion en imposant certaines restrictions. Enfin nous concluons dans le dernier paragraphe en présentant quelques perspectives pour des travaux futurs.

Relation	Symbole	Inverse	Illustration
x before y	b	bi	
x meets y	m	mi	
x overlaps y	o	oi	
x starts y	s	si	
x during y	d	di	
x finishes y	f	fi	
x equal y	eq	eq	

FIGURE 1 – L'ensemble \mathcal{B}_{int} des 13 relations de base de l'algèbre des intervalles.

2 Rappels sur les algèbres qualitatives et les RCQ

Une algèbre qualitative considère un ensemble fini \mathcal{B} de relations binaires appelées *relations de base* sur un domaine \mathcal{D} . Chacune des relations de base de \mathcal{B} représente une situation qualitative particulière entre deux entités prenant leur valeurs dans \mathcal{D} . Ces relations sont supposées complètes et mutuellement exclusives, autrement dit deux éléments de \mathcal{D} satisfont une et une seule relation de base de \mathcal{B} . $2^{\mathcal{B}}$ dénote l'ensemble de tous les sous-ensembles de \mathcal{B} . Les éléments de $2^{\mathcal{B}}$ sont appelés *relations*. Une relation $R \in 2^{\mathcal{B}}$ représente l'ensemble de relations de base possibles entre deux éléments de \mathcal{D} . L'une des relations de base est la relation identité, notée *eq*. Chacune des relations de base r de \mathcal{B} est associée à une relation inverse $r^{-1} \in \mathcal{B}$ telle que $\forall X, Y \in \mathcal{D}, XrY \text{ ssi } Yr^{-1}X$.

Par exemple, la figure 1 décrit l'ensemble \mathcal{B}_{int} des 13 relations de base de l'algèbre d'Allen (ou algèbre des intervalles) pouvant être satisfaites entre deux intervalles de la droite des rationnels. Nous avons $\mathcal{B}_{int} = \{eq, b, bi, m, mi, o, oi, s, si, d, di, f, fi\}$.

Un réseau de contraintes qualitatives (RCQ en abrégé) est utilisé pour représenter un ensemble de configurations qualitatives possibles entre des entités. Etant donné un ensemble de relations de base \mathcal{B} , un RCQ N défini sur $2^{\mathcal{B}}$ est une paire (V, C) où $V = \{v_1, \dots, v_n\}$ est un ensemble de variables représentant les entités spatiales ou temporelles considérées, et C est une application qui associe à chaque paire de variables (v_i, v_j) une relation de $2^{\mathcal{B}}$. $C(v_i, v_j)$ est également appelée *contrainte*. Nous préférons la notation C_{ij} plutôt que $C(v_i, v_j)$ par souci de concision. Pour tout $v_i, v_j \in V$, $C_{ji} = \{r \mid r^{-1} \in C_{ij}\}$ et $C_{ii} = \{eq\}$.

Nous introduisons maintenant quelques définitions sur les RCQ. Soit \mathcal{B} un ensemble de relations de base

et $N = (V, C)$ un RCQ défini sur $2^{\mathcal{B}}$. Une *instanciation cohérente* de N sur $V' \subseteq V$ est une application α de V' dans \mathcal{D} telle que $\alpha(v_i) C_{ij} \alpha(v_j)$, pour tout $v_i, v_j \in V'$. N est dit *cohérent* ssi il existe une instanciation cohérente de N sur V . Un *sous-réseau* de N est un RCQ $N' = (V, C')$ où $C'_{ij} \subseteq C_{ij}$ pour tout $v_i, v_j \in V$. Un *scénario cohérent* de N est un sous-réseau cohérent de N dans lequel chaque contrainte est composée d'une et une seule relation de base de \mathcal{B} (une telle contrainte est alors appelée contrainte atomique).

$[N]$ représente l'ensemble des scénarios cohérents de N . Nous notons N_{ALL}^V le RCQ (V, C) tel que pour tout $v_i, v_j \in V$, $C_{ij} = \mathcal{B}$ si $v_i \neq v_j$ et $C_{ij} = \{eq\}$ sinon.

3 Fusion de RCQ

Etant donné un ensemble de relations de base \mathcal{B} , nous considérons un ensemble $\mathcal{N} = \{N_1, \dots, N_m\}$ de RCQ définis sur $2^{\mathcal{B}}$ et sur un ensemble commun de variables $V = \{v_1, \dots, v_n\}$.

Exemple 1. *Considérons les trois RCQ définis sur $2^{\mathcal{B}_{int}}$ et sur $V = \{v_1, v_2, v_3, v_4\}$, schématisés sur la figure 2. Nous utilisons les conventions suivantes pour la représentation schématique d'un RCQ : pour tout v_i, v_j , nous ne représentons ni les contraintes $C_{ii} = \{eq\}$, ni les contraintes C_{ji} si C_{ij} est représentée, puisque $C_{ji} = \{r \mid r^{-1} \in C_{ij}\}$. Enfin lorsqu'un RCQ ne fournit aucune information entre deux variables (par exemple, la paire (v_2, v_3) pour le RCQ N_1), la contrainte associée correspond implicitement à l'ensemble \mathcal{B} et elle n'est pas représentée.*

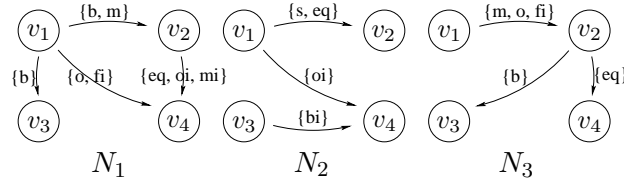


FIGURE 2 – Trois RCQ à fusionner.

Notre but est d'obtenir un ensemble cohérent d'informations représentant de manière globale l'ensemble des RCQ de \mathcal{N} . La manière la plus naturelle de répondre à ce but est de prendre comme résultat l'ensemble de tous les scénarios cohérents admis par chacun des RCQ de \mathcal{N} , c'est-à-dire l'ensemble $\bigcap_{N_i \in \mathcal{N}} [N_i]$. Cependant cet ensemble peut être vide. En effet, considérons l'exemple ci-dessus. Chacun des scénarios appartenant à l'ensemble $[N_1]$ admet une des relations de base de $\{b, m\}$ entre les variables v_1 et v_2 , et chacun des scénarios de $[N_2]$ admet une des relations de base de $\{s, eq\}$ entre ces mêmes variables. Il

n'existe pas de scénario cohérent commun à $[N_1]$ et $[N_2]$, et donc $\bigcap_{N_i \in \mathcal{N}} [N_i] = \emptyset$.

Condotta et al. [5] ont proposé une méthode parcimonieuse pour la fusion de RCQ, qui retourne en résultat un ensemble non vide de scénarios cohérents les plus proches (en termes de distance) de l'ensemble \mathcal{N} . Dans cette approche, l'algorithme de fusion nécessite le calcul de distances à propos de tous les scénarios cohérents possibles sur V , donc appartenant à l'ensemble $[N_{ALL}^V]$. En outre l'ensemble de scénarios cohérents résultants peut être de taille exponentielle. Ces deux critères rendent cette approche difficile à mettre en œuvre.

Nous décrivons dans cet article une approche pratique pour la fusion de RCQ. Cette approche fusionne indépendamment les contraintes portant sur les mêmes variables. Notre méthode est basée sur l'algorithme suivant. Pour chaque paire de variables (v_i, v_j) nous appliquons un opérateur de fusion sur l'ensemble de toutes les contraintes des RCQ entre v_i et v_j . Nous obtenons alors une nouvelle contrainte correspondant à l'ensemble des relations de base de \mathcal{B} représentant une information globale pour la contrainte entre v_i et v_j . Nous obtenons un nouvel ensemble de contraintes C , chacune représentant une vision globale des contraintes correspondantes de chaque RCQ à fusionner. Nous définissons alors le résultat de la fusion comme étant le RCQ $N_{res} = (V, C)$.

4 Un nouvel algorithme pour la fusion de RCQ

4.1 Fusion de contraintes

Rappelons que notre but est de fusionner un ensemble de RCQ $\mathcal{N} = \{N_1, \dots, N_m\}$ définis sur $2^{\mathcal{B}}$ et sur le même ensemble de variables V . L'approche consiste à fusionner indépendamment les contraintes pour chaque paire de variables (v_i, v_j) de V en utilisant un opérateur de fusion ϑ pour les relations de $2^{\mathcal{B}}$. Par conséquent notre approche nécessite la définition d'un opérateur de fusion de contraintes ϑ .

Définition 1 (opérateur de fusion de contraintes). *Soit $\mu \in 2^{\mathcal{B}}$ une relation appelée relation d'intégrité. ϑ est un opérateur qui associe à une relation d'intégrité μ et un multi-ensemble de relations de $2^{\mathcal{B}}$ noté \mathcal{R} , une relation de $2^{\mathcal{B}}$ notée R . ϑ est un opérateur de fusion de contraintes s'il satisfait au moins les propriétés suivantes, $\forall \mu \in 2^{\mathcal{B}}, \forall \mathcal{R}$ ensemble de relations de $2^{\mathcal{B}}$:*

- (a) $\vartheta(\mu, \mathcal{R}) \subseteq \mu$,
- (b) si $\mu \neq \emptyset$, alors $\vartheta(\mu, \mathcal{R}) \neq \emptyset$,
- (c) si $\mu \cap \bigcap_{R \in \mathcal{R}} R \neq \emptyset$, alors $\vartheta(\mu, \mathcal{R}) = \mu \cap \bigcap_{R \in \mathcal{R}} R$.

La propriété (a) assure que le résultat de la fusion est une relation dont les relations de base appartiennent à la relation d'intégrité μ . La propriété (b) garantit que si μ n'est pas la relation vide alors le résultat de la fusion n'est pas la relation vide. La propriété (c) exige que $\vartheta(\mu, \mathcal{R})$ soit l'ensemble des relations de base appartenant à μ et à chacune des relations de \mathcal{R} , si cet ensemble n'est pas vide.

Soit \mathcal{N} un ensemble de RCQ définis sur $2^{\mathcal{B}}$ et V . Notons $V_{<}^2$ l'ensemble $\{(v_i, v_j) \in V \times V \mid i < j\}$. Pour chaque paire de variables (v_i, v_j) de $V_{<}^2$, nous désignons par \mathcal{C}_{ij} l'ensemble des contraintes des RCQ de \mathcal{N} portant sur la paire de variable (v_i, v_j) . L'algorithme 1 décrit la construction par morceaux du RCQ résultat $N_{res} = (V, C)$.

Algorithme 1 : Fusion par morceaux d'un ensemble de RCQ

Entrée : un ensemble \mathcal{N} de RCQ définis sur $2^{\mathcal{B}}$ et V ,
un opérateur de fusion de contraintes ϑ

```

1 début
2    $N_{res} = (V, C);$ 
3    $\mu = \mathcal{B};$ 
4   pour tout  $(v_i, v_j) \in V_{<}^2$  faire
5      $C_{ij} = \vartheta(\mu, \mathcal{C}_{ij});$ 
6   fin faire
7   retourner  $N_{res};$ 
8 fin
```

Nous pouvons observer que pour tout ensemble \mathcal{C}_{ij} à fusionner nous n'imposons aucune restriction sur μ , le sur-ensemble du résultat devant être renvoyé par l'opérateur ϑ (cf. ligne 3 de l'algorithme). Ceci est dû au fait que nous ne considérons pas les dépendances entre les contraintes résultantes du RCQ N_{res} pour l'instant.

Notons que puisque nous obtenons en résultat un RCQ, la complexité spatiale de cette méthode, linéaire en la taille de l'entrée, est beaucoup plus intéressante que celle proposée dans [5], dans laquelle le résultat renvoyé est de taille exponentielle. Sa complexité temporelle dépend toutefois de la construction de ϑ .

Pour définir un opérateur de fusion de contraintes ϑ au sens de la définition 1, nous exploitons les travaux effectués dans [9] pour la fusion d'opérateurs de fusion de bases propositionnelles basés sur un calcul de distances. Le principal ingrédient utilisé dans de tels opérateurs de fusion est une « distance », ce qui permet d'obtenir comme résultat un ensemble de connaissances les plus « proches » de l'ensemble à fusionner. Dans le cadre de la logique propositionnelle, étant donné un ensemble \mathcal{K} de bases de croyances pro-

positionnelles à fusionner et une formule propositionnelle IC (pour *contrainte d'intégrité*), le résultat de la fusion correspond à l'ensemble des modèles de IC qui sont les plus proches (en termes de distance) de l'ensemble \mathcal{K} .

Dans le même esprit, étant donné un ensemble $\mathcal{R} = \{R_1, \dots, R_p\}$ de relations de $2^{\mathcal{B}}$ et une relation $\mu \subseteq \mathcal{B}$ représentant une relation d'intégrité, on peut définir un opérateur de fusion de contraintes en trois étapes.

Tout d'abord, on calcule une distance locale entre chacune des relations de base de μ et chaque relation $R_i \in \mathcal{R}, i \in \{1, \dots, p\}$. La distance entre une relation de base $r_1 \in \mu$ et une relation R est la distance minimale entre r_1 et chacune des relations de base de R . Formellement,

$$d(r_1, R) = \begin{cases} \min\{d_r(r_1, r_2) \mid r_2 \in R\} & \text{si } R \neq \emptyset \\ 0 & \text{sinon.} \end{cases}$$

Nous devons donc disposer d'une distance locale entre relations de base de \mathcal{B} .

Définition 2 (distance entre relations de base).

Une distance d_r entre deux relations de base de \mathcal{B} est une pseudo-distance, c'est-à-dire une application de $\mathcal{B} \times \mathcal{B}$ dans \mathbb{R}^+ telle que $\forall r_1, r_2 \in \mathcal{B}$,

$$\begin{cases} d_r(r_1, r_2) = d_r(r_2, r_1) \\ d_r(r_1, r_2) = 0 \text{ ssi } r_1 = r_2 \\ d_r(r_1, r_2) = d_r(r_1^{-1}, r_2^{-1}), \end{cases}$$

où r_1^{-1} (resp. r_2^{-1}) est la relation inverse de r_1 (resp. r_2).

Par exemple, pour la distance drastique, la distance entre deux relations de base est égale à 1 si elles sont différentes, 0 sinon.

Dans le but de définir une distance locale d_r plus pertinente, il est important de rappeler qu'une algèbre qualitative incorpore un concept de voisinage entre relations de base. Ce voisinage est usuellement représenté par un treillis sur l'ensemble des relations de base de \mathcal{B} . Pour de nombreux formalismes qualitatifs un tel treillis existe [11, 12]. Ce treillis permet alors de déterminer un graphe de voisinage conceptuel entre relations de base. La figure 3 schématise le graphe de voisinage entre les relations de base de \mathcal{B}_{int} , correspondant au diagramme de Hasse du treillis des intervalles [11]. La distance de voisinage conceptuel entre relations de base [5] exploite ce graphe et est définie comme suit :

Définition 3 (distance de voisinage conceptuel).

Soit $\mathcal{G}_{\mathcal{B}}$ le diagramme de Hasse du treillis sur \mathcal{B} . La distance de voisinage conceptuel entre deux relations de base r et r' , notée $d_v(r, r')$ est la longueur de la chaîne la plus courte dans $\mathcal{G}_{\mathcal{B}}$ reliant r à r' .

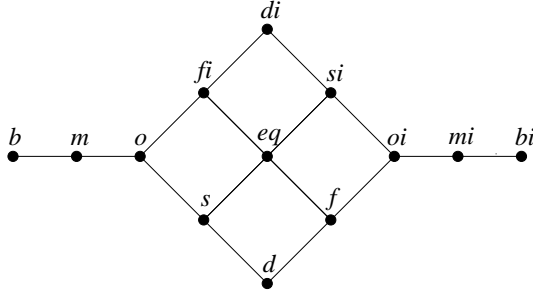


FIGURE 3 – Le graphe de voisinage conceptuel de l'algèbre des intervalles.

Par exemple, pour des relations de base de \mathcal{B}_{int} , on obtient $d_v(di, f) = 3$ puisque (di, fi, eq, f) est l'une des chaînes les plus courtes entre di et f dans le graphe de voisinage associé (cf. Figure 3).

La deuxième étape consiste à agréger les distances locales calculées à l'étape précédente afin d'obtenir une distance globale entre une relation de base r et l'ensemble \mathcal{R} , en utilisant une fonction d'agrégation.

Définition 4 (fonction d'agrégation). Une fonction d'agrégation est une application \otimes qui associe à un ensemble de nombres réels positifs un nombre réel positif, et qui satisfait les propriétés suivantes :

- si $x_1 \leq x'_1, \dots, x_n \leq x'_n$, alors $\otimes(x_1, \dots, x_n) \leq \otimes(x'_1, \dots, x'_n)$ (monotonie)
- si $x_1 = \dots = x_n = 0$, alors $\otimes(x_1, \dots, x_n) = 0$ (minimalité)

Nous considérons une propriété supplémentaire pour une fonction d'agrégation \otimes :

Définition 5 (minimalité forte). Une fonction d'agrégation \otimes satisfait le postulat de minimalité forte ssi

$$si \otimes(x_1, \dots, x_n) = 0, \text{ alors } x_1 = \dots = x_n = 0.$$

Plusieurs fonctions d'agrégation ont été étudiées dans la littérature [13, 16]. Par exemple, la fonction de majorité \sum [13] qui effectue la somme de ses arguments favorise le point de vue de la majorité des sources. La fonction d'arbitrage \mathcal{MAX} [16] qui retourne quant à elle l'argument de valeur maximale sera plus appropriée si l'on désire minimiser l'insatisfaction maximale sur l'ensemble des sources par rapport au résultat. Etant donné une fonction d'agrégation, on calcule la distance globale d_\otimes entre une relation de base r et l'ensemble \mathcal{R} . Cette distance globale est définie comme suit :

$$d_\otimes(r, \mathcal{R}) = \otimes\{d(r, R) \mid R \in \mathcal{R}\}.$$

Enfin, le résultat de la fusion, noté $\vartheta^{d_r, \otimes}(\mu, \mathcal{R})$, est le sous-ensemble des relations de base de μ les plus « proches » de l'ensemble \mathcal{R} . Formellement,

$$\vartheta^{d_r, \otimes}(\mu, \mathcal{R}) = \{r \in \mu \mid \nexists r' \in \mu : d_\otimes(r', \mathcal{R}) < d_\otimes(r, \mathcal{R})\}.$$

Proposition 1. Si \otimes est une fonction d'agrégation vérifiant le postulat de minimalité forte, alors $\vartheta^{d_r, \otimes}$ est un opérateur de fusion de contraintes, au sens de la définition 1.

Nous donnons les preuves de quelques propositions en annexe.

Remarquons que les fonctions d'agrégation \sum et \mathcal{MAX} satisfont le postulat de minimalité forte. Il en résulte que $\vartheta^{d_r, \sum}$ et $\vartheta^{d_r, \mathcal{MAX}}$ sont des opérateurs de fusion de contraintes, au sens de la définition 1.

Exemple 1 (suite). Considérons de nouveau les trois RCQ de l'exemple (cf. Figure 2). Soit $\mathcal{C}_{12} = \{\{b, m\}, \{s, eq\}, \{m, o, fi\}\}$ l'ensemble des contraintes des RCQ portant sur la paire de variables (v_1, v_2) , et soit $\mu = \mathcal{B}$. Pour chaque relation de base $r \in \mu$ on calcule une distance locale entre r et chaque relation de \mathcal{C}_{12} . Par exemple, en utilisant la distance de voisinage conceptuel entre relations de base, la distance locale entre la relation de base b et la relation $\{m, o, fi\}$ est donnée par

$$\begin{aligned} d(b, \{m, o, fi\}) &= \min\{d_v(b, m), d_v(b, o), d_v(b, fi)\} \\ &= \min\{1, 2, 3\} = 1. \end{aligned}$$

En utilisant la fonction d'agrégation $\otimes = \sum$, on obtient

$$d_\sum(d(b, \{b, m\}), d(b, \{s, eq\}), d(b, \{m, o, fi\})) = 4.$$

Le résultat de la fusion $\vartheta^{d_v, \sum}(\mu, \mathcal{C}_{12})$ correspond aux relations de base de μ les plus proches de l'ensemble des relations de \mathcal{C}_{12} , c'est-à-dire l'ensemble $\{m, o\}$.

Évaluons maintenant la complexité de calcul de $\vartheta^{d_r, \otimes}(\mu, \mathcal{R})$. Notons $|E|$ le nombre d'éléments d'un ensemble fini E . $f(\otimes, |\mathcal{R}|)$, respectivement $f(d_r)$ correspond au nombre maximum d'étapes du calcul de \otimes sur ses $|\mathcal{R}|$ arguments, respectivement d_r . Comme d_r est une distance entre relations de base d'un ensemble donné \mathcal{B} , on a $f(d_r) \in O(1)$. La distance d entre une relation de base et une relation se calcule alors également en temps constant, puisque $|\mathcal{R}|$ est borné par une constante. Donc le calcul de $\vartheta^{d_r, \otimes}(\mu, \mathcal{R})$ s'effectue en $\alpha \cdot f(\otimes, |\mathcal{R}|)$ étapes de calcul pour une certaine constante α . En outre, la plupart des fonctions d'agrégation « usuelles » sont calculables en temps polynomial, et donc dans ce cas la fusion de contraintes s'effectue en temps polynomial. Par exemple, si $\otimes \in \{\sum, \mathcal{MAX}\}$, on a $f(\otimes, |\mathcal{R}|) \in O(|\mathcal{R}|)$. Dans ce cas $\vartheta^{d_r, \otimes}(\mu, \mathcal{R})$ est calculable en temps $O(|\mathcal{R}|)$.

Ces résultats se détachent considérablement de la fusion de bases propositionnelles qui est typiquement un problème NP-difficile ; par contre des résultats de complexité similaires pourraient être obtenus en limitant le nombre de variables propositionnelles à une certaine constante donnée.

4.2 Maintenir la cohérence du RCQ résultant

Nous n'avons aucune garantie à propos de la cohérence du RCQ résultant de la fusion N_{res} . Dans le but de résoudre ce problème, notre approche est basée sur l'idée suivante : nous construisons l'ensemble des contraintes résultantes C étape par étape. A l'initialisation du processus, chacune des contraintes de C est définie par l'ensemble \mathcal{B} (autrement dit N_{res} est initialisé au RCQ N_{ALL}^V). A chaque étape, nous choisissons une paire (v_i, v_j) dans $V_{<}^2$, et nous associons à la contrainte C_{ij} le résultat de la fusion de l'ensemble C_{ij} tout en préservant la cohérence du RCQ.

Pour cela, nous supposons l'existence d'un ordre total \prec sur $V_{<}^2$ dans le but de résoudre le problème du choix de la contrainte à chaque étape : les paires de variables sont alors sélectionnées les unes après les autres en respectant l'ordre \prec .

Cet ordre peut être obtenu de différentes manières. Par exemple toutes les paires de variables de $V_{<}^2$ peuvent être ordonnées en fonction d'un certain degré de priorité ou de fiabilité, lui-même résultant de l'agrégation de degrés locaux définis sur chaque RCQ. Par défaut, cet ordre est défini comme étant l'ordre lexicographique sur les indices des variables.

Le processus global de fusion est donné dans l'algorithme 2.

L'algorithme initialise tout d'abord chaque contrainte du RCQ résultant N_{res} à la relation universelle \mathcal{B} (lignes 3 à 5). Puis il sélectionne la prochaine paire (v_i, v_j) de variables de $V_{<}^2$ en respectant l'ordre \prec (cette paire correspond à la prochaine contrainte C_{ij} à affecter, ligne 6). Des lignes 7 à 12, SAT est un test de cohérence du RCQ donné en paramètre. Notons $N[(i, j, r)]$ le RCQ $N = (V, C)$ dans lequel on assigne à la contrainte C_{ij} la relation de base r . On définit la relation d'intégrité μ_{ij} comme étant l'ensemble des relations de base $r \in \mathcal{B}$ pour lesquelles $N_{res}[(i, j, r)]$ est cohérent. Ensuite, la contrainte courante C_{ij} est définie comme étant le résultat de la fusion de C_{ij} sous la relation d'intégrité μ_{ij} , ligne 13. L'algorithme se termine lorsque toutes les contraintes de N_{res} ont été assignées.

Dès lors que le problème SAT est un problème NP-difficile, même si ϑ est calculable en temps polynomial, l'algorithme 2 ne calcule pas le RCQ résultant en temps polynomial. En effet il effectue un nombre d'appels linéaire à SAT (linéaire en nombre de contraintes

Algorithme 2 : Fusion cohérente d'un ensemble de RCQ

Entrée : un ensemble $\mathcal{N} = \{N_1, \dots, N_m\}$ de RCQ sur V ,
un opérateur de fusion de contraintes ϑ ,
un ordre total \prec sur l'ensemble $V_{<}^2$
Sortie : un RCQ $N_{res} = (V, C)$

```

1 début
2    $N_{res} = (V, C)$ ;
3   pour tout  $(v_i, v_j) \in V_{<}^2$  faire
4      $C_{ij} = \mathcal{B}$ ;
5   fin faire
6   pour tout  $(v_i, v_j) \in V_{<}^2$  dans l'ordre  $\prec$  faire
7      $\mu_{ij} = \emptyset$ ;
8     pour tout  $r \in \mathcal{B}$  faire
9       si  $SAT(N_{res}[(i, j, r)])$  alors
10         $\mu_{ij} = \mu_{ij} \cup \{r\}$ ;
11      fin si
12    fin faire
13     $C_{ij} = \vartheta(\mu_{ij}, C_{ij})$ ;
14  fin faire
15  retourner  $N_{res}$ ;
16 fin

```

$|V_{<}^2|$ à traiter).

L'algorithme 2 donne lieu à la proposition suivante :

Proposition 2. N_{res} est cohérent.

La proposition suivante exprime le fait que fusionner localement les contraintes des RCQ conduit à une propriété intéressante à propos du RCQ final : si les RCQ donnés en entrée ne sont pas conflictuels, alors l'algorithme 2 retourne en résultat un RCQ dont l'ensemble des scénarios cohérents correspond exactement à l'ensemble des scénarios cohérents communs à chacun des RCQ donnés en entrée.

Proposition 3.

$$\bigcap_{N_i \in \mathcal{N}} [N_i] \neq \emptyset \Rightarrow [N_{res}] = \bigcap_{N_i \in \mathcal{N}} [N_i].$$

Exemple 1 (suite). Nous appliquons l'algorithme 2 pour fusionner l'ensemble des trois RCQ $\mathcal{N} = \{N_1, N_2, N_3\}$ de notre même exemple (cf. Figure 2). Pour $\vartheta^{d_r, \otimes}$ nous choisissons $d_r = d_v$, la distance de voisinage conceptuel entre relations de base, et $\otimes = \sum$. Nous définissons l'ordre \prec comme étant l'ordre lexicographique sur les indices des variables. La figure 4 résume le processus complet qui consiste à affecter étape par étape chaque contrainte du RCQ résultant N_{res} en respectant l'ordre \prec . N_{res} est le RCQ cohérent obtenu après l'étape 6. Chaque contrainte de N_{res} est composée des relations de base les plus proches de l'ensemble

des contraintes correspondantes dans les RCQ de \mathcal{N} de telle sorte qu'à chaque étape N_{res} est maintenu cohérent.

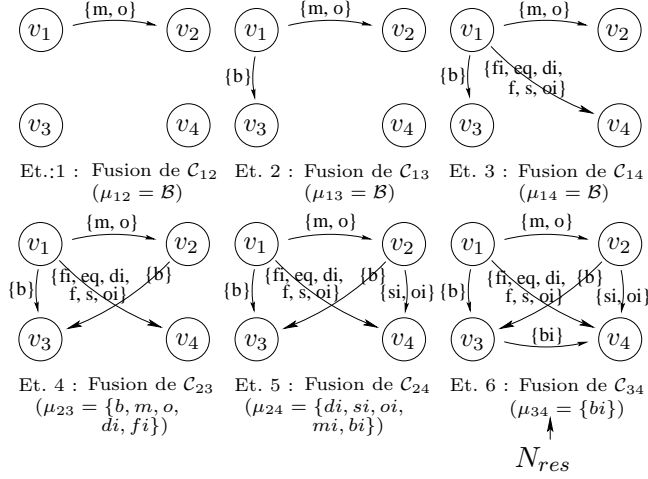


FIGURE 4 – Le processus de fusion, étape par étape.

5 Amélioration de l'algorithme

5.1 Eviter des tests de cohérence inutiles

Afin d'affecter chacune des contraintes de N_{res} , l'algorithme 2 effectue à chaque fois un nombre constant d'appels à la fonction SAT (lignes 7 à 12). Notre but est maintenant d'améliorer l'algorithme en supprimant les appels inutiles à SAT pour certaines contraintes.

Dans cette section, nous considérerons les algèbres qualitatives vérifiant la propriété suivante : soit deux RCQ cohérents $N_1 = (V_1, C_1)$ et $N_2 = (V_2, C_2)$ définis sur $2^{\mathcal{B}}$, et respectivement définis sur deux ensembles disjoints de variables, et soit le RCQ N défini sur $V = V_1 \cup V_2$ tel que N_1 et N_2 en sont des sous-réseaux. Pour tout $v_i \in V_1$, $v_j \in V_2$, $r \in \mathcal{B}$, le RCQ $N[(i, j, r)]$ conserve sa cohérence. Cette propriété est vérifiée pour la plupart des algèbres qualitatives définies dans la littérature, en particulier toutes celles dont les relations considérées sont basées sur une relation d'ordre [1, 3, 11, 12]. A notre connaissance, cette propriété reste indémontrée dans le cas des relations de type topologiques [15].

Etant donné un RCQ $N = (V, C)$ sur $2^{\mathcal{B}}$, nous dirons que C_{ij} est *inactive* si $C_{ij} = \mathcal{B}$, *active* sinon.

Proposition 4. *Soit $N = (V, C)$ un RCQ cohérent sur \mathcal{B} et (v_i, v_j) une paire de variables de V . S'il n'existe pas de chaîne de contraintes actives de N C_{ik}, \dots, C_{pj} , alors $\forall r \in \mathcal{B}$, $N[(i, j, r)]$ est cohérent.*

Supposons que pendant l'exécution de l'algorithme 2, la contrainte à affecter est C_{ij} . D'après la proposi-

tion 4 nous pouvons remplacer les lignes 7 à 12 par les suivantes :

```

si il n'existe pas de chaîne de contraintes actives
dans  $N_{res}$  entre  $v_i$  et  $v_j$  alors
  |  $\mu_{ij} = \mathcal{B}$ ;
sinon
  | // lignes 7 à 12 de l'algorithme 2
fin

```

Soit \mathcal{N} un ensemble de RCQ définis sur V . Si pour une paire (v_i, v_j) de V toutes les contraintes de C_{ij} sont définies par l'ensemble \mathcal{B} (autrement dit, aucune des sources n'a d'information à propos de la contrainte entre les variables v_i et v_j), alors nous dirons qu'une telle paire est *non contrainte par rapport à \mathcal{N}* , *contrainte par rapport à \mathcal{N}* sinon.

Définition 6 (arbre-ensemble). *Soit \mathcal{N} un ensemble de RCQ définis sur V . \mathcal{N} est un arbre-ensemble s'il n'existe pas de cycle de paires de variables de V contraintes par rapport à \mathcal{N} .*

Proposition 5. *Soit \mathcal{N} un ensemble de RCQ définis sur V , ϑ un opérateur de fusion de contraintes, et \prec un ordre quelconque sur V_{\prec}^2 . Si \mathcal{N} est un arbre-ensemble, alors pour toute paire (v_i, v_j) de V contrainte par rapport à \mathcal{N} , $\forall r \in \mathcal{B}$, $N_{res}[(i, j, r)]$ est cohérent.*

Par conséquent, si \mathcal{N} est un arbre-ensemble de RCQ, les lignes 7 à 12 de l'algorithme 2 peuvent alors simplement être remplacées par l'instruction $\mu_{ij} = \mathcal{B}$. Nous pouvons alors en déduire le lemme suivant :

Lemme 1. *Si \mathcal{N} est un arbre-ensemble, alors l'algorithme 2 s'exécute en temps polynomial. De plus, le RCQ résultant N_{res} est indépendant de l'ordre \prec sur V_{\prec}^2 choisi.*

5.2 Exploitation des classes traitables

En général le problème de la cohérence d'un RCQ est un problème NP-difficile. Dans le but de pallier ce problème de nombreux fragments traitables ont été identifiés dans différentes algèbres qualitatives [14, 12]. Etant donnée une algèbre qualitative définie sur l'ensemble de relations de base \mathcal{B} , considérons un sous-ensemble \mathcal{S} de $2^{\mathcal{B}}$. \mathcal{S} est qualifié de sous-ensemble traitable ssi la cohérence de tout RCQ dont les contraintes sont définies par un élément de \mathcal{S} peut être décidée en temps polynomial. Dans notre cadre, \mathcal{S} doit être nécessairement fermée pour l'intersection (\cap), définie comme l'intersection ensembliste usuelle, et l'inverse ($^{-1}$), définie pour tout $R \in 2^{\mathcal{B}}$ comme étant la relation $R^{-1} = \{r \mid r^{-1} \in R\}$. De plus nous imposons le fait que \mathcal{S} contienne la relation universelle \mathcal{B} et la relation atomique $\{eq\}$.

Lorsqu'un tel ensemble traitable \mathcal{S} est identifié, l'algorithme 2 peut être amélioré en assignant chacune des contraintes du RCQ construit N_{res} à une relation de \mathcal{S} . En procédant ainsi tous les tests de cohérence (ligne 9 de l'algorithme) peuvent être effectués en temps polynomial, et ainsi l'algorithme 2 peut s'exécuter en temps polynomial.

Dans cette optique, pour chaque paire $(v_i, v_j) \in V_{<}^2$, nous calculons $\vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$ et nous assignons alors la contrainte résultante C_{ij} à la relation de \mathcal{S} la plus « proche » de $\vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$. Il est donc nécessaire de disposer d'une distance $d_R(R_1, R_2)$ entre deux relations de $2^{\mathcal{B}}$.

Définition 7 (distance entre relations). Une distance d_R entre deux relations de $2^{\mathcal{B}}$ est une pseudo-distance, c'est-à-dire une application de $2^{\mathcal{B}} \times 2^{\mathcal{B}}$ dans \mathbb{R}^+ telle que $\forall R_1, R_2 \in 2^{\mathcal{B}}$,

$$\begin{cases} d_R(R_1, R_2) = d_R(R_2, R_1) \\ d_R(R_1, R_2) = 0 \text{ ssi } R_1 = R_2 \end{cases}$$

Nous considérons une propriété supplémentaire pour une distance d_R entre relations :

Définition 8 (inégalité stricte). Une distance d_R entre relations satisfait le postulat d'inégalité stricte ssi $\forall R_1, R_2, R_3 \in 2^{\mathcal{B}}$,

$$R_1 \subset R_2 \subset R_3 \Rightarrow d_R(R_2, R_3) < d_R(R_1, R_3).$$

Dans cette section, nous imposons que la distance entre relations utilisée vérifie le postulat d'inégalité stricte. Cette restriction reste faible. En effet, plus généralement, la plupart des distances entre ensembles d'éléments quelconques satisfont ce postulat. Une distance entre relations peut être définie de différentes manières, nous rappelons l'une des plus naturelles, la distance euclidienne.

Définition 9 (distance euclidienne entre relations). Soit d une distance locale entre une relation de base de \mathcal{B} et une relation de $2^{\mathcal{B}}$. La distance euclidienne entre deux relations de $2^{\mathcal{B}}$ est définie comme suit, $\forall R_1, R_2 \in 2^{\mathcal{B}}$:

$$d_{Euc}(R_1, R_2) = \sqrt{\sum_{r \in \mathcal{B}} (d(r, R_1) - d(r, R_2))^2}.$$

Soit $N_{res}^* = (V, C)$ le RCQ résultant de la fusion. Pour chaque paire $(v_i, v_j) \in V_{<}^2$, la contrainte C_{ij} est assignée à la relation $R_{ij}^* \in \mathcal{S}$ la plus « proche » de $\vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$ par rapport à d_R , avec $R_{ij}^* \supseteq \vartheta^{d_r, \otimes}(\mathcal{C}_{ij})$. Formellement, pour toute paire (v_i, v_j) de $V_{<}^2$, R_{ij}^* est la relation de \mathcal{S} telle que $R_{ij}^* \supseteq \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$, et $\nexists R \in \mathcal{S}$ telle que $R \supseteq \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$, $d_R(R, \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})) < d_R(R_{ij}^*, \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij}))$.

Proposition 6. $\forall (v_i, v_j) \in V_{<}^2$, R_{ij}^* existe et est unique.

Par conséquent nous remplaçons la ligne 13 de l'algorithme 2 par l'instruction $C_{ij} = R_{ij}^*$.

Puisque pour chaque paire $(v_i, v_j) \in V_{<}^2$ nous avons $R_{ij}^* \supseteq \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$, chaque contrainte C_{ij} est instanciée par une relation qui contient toutes les relations de base de $\vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$. Donc d'après la proposition 2, nous avons la proposition suivante :

Proposition 7. N_{res}^* est cohérent.

Si les RCQ en entrée ne sont pas conflictuels, l'ensemble des scénarios cohérents de N_{res}^* ne coïncide pas nécessairement avec l'ensemble des scénarios communs à chacun des RCQ en entrée, comme c'est le cas en utilisant la méthode générale (cf. Proposition 3). Néanmoins nous avons le résultat suivant, plus faible :

Proposition 8. Si toutes les contraintes des RCQ de \mathcal{N} sont définies par des relations de \mathcal{S} , alors

$$\bigcap_{N_i \in \mathcal{N}} [N_i] \neq \emptyset \Rightarrow [N_{res}^*] = \bigcap_{N_i \in \mathcal{N}} [N_i].$$

6 Conclusion

Nous avons proposé dans cet article une nouvelle méthode efficace pour la fusion de réseaux de contraintes qualitatives (RCQ) définis sur une algèbre qualitative commune. Notre méthode est générique puisque qu'elle capture la plupart des algèbres qualitatives définies dans la littérature. Etant donné un ensemble de RCQ éventuellement conflictuels, le processus de fusion retourne un RCQ cohérent représentant une vision globale des RCQ en entrée. Il consiste à fusionner localement les contraintes des RCQ. En s'inspirant des travaux effectués dans le cadre de la fusion de bases propositionnelles, nous avons défini une classe d'opérateurs de fusion de contraintes basés sur un calcul de distance qui permet de fusionner localement les contraintes des RCQ en entrée pour construire pas à pas les contraintes du RCQ résultant. Nous avons proposé un compromis intéressant entre efficacité du nouveau processus et propriétés logiques sur le RCQ résultant, et nous avons également montré comment obtenir un algorithme de fusion en temps polynomial en imposant certaines restrictions.

Ce travail peut être étendu dans plusieurs directions. La définition de l'ordre $<$ n'a aucune influence sur le RCQ résultant de la fusion lorsque l'ensemble de RCQ à fusionner est un arbre-ensemble de RCQ (cf. paragraphe 5). Cependant cette notion d'arbre-ensemble de RCQ peut paraître assez restrictive. Une problématique de recherche future est de caractériser de manière

plus faible la structure de l'ensemble de RCQ à fusionner et d'étudier comment on peut réduire l'influence de la définition de l'ordre \prec sur le RCQ résultant.

Konieczny et al. [10] ont proposé une caractérisation logique des opérateurs de fusion propositionnelle avec contraintes d'intégrité sous la forme d'un ensemble de postulats de rationalité. Un travail futur sera d'inclure une extension de ces postulats dans le contexte des opérateurs de fusion de contraintes.

Références

- [1] J-F. Allen. An interval-based representation of temporal knowledge. In *IJCAI*, pages 221–226, 1981.
- [2] J-F. Allen. Towards a general theory of action and time. *Artificial Intelligence*, 23(2) :123–154, 1984.
- [3] P. Balbiani and J-F. Condotta. Spatial reasoning about points in a multidimensional setting. *Applied Intelligence*, 17(3) :221–238, 2002.
- [4] P. Balbiani and A. Osmani. A model for reasoning about topological relations between cyclic intervals. In *KR*, pages 378–385, 2000.
- [5] J-F. Condotta, S. Kaci, and N. Schwind. A framework for merging qualitative constraints networks. In *FLAIRS*, pages 586–591, 2008.
- [6] A. Gerevini and J. Renz. Combining topological and size information for spatial reasoning. *Artificial Intelligence*, 137 :1–42, 2002.
- [7] I. Hamlet and J. Hunter. A representation of time for medical expert systems. In *AIME*, pages 112–119, 1987.
- [8] H. A. Kautz. *A formal theory of plan recognition*. PhD thesis, Rochester, NY, USA, 1987.
- [9] S. Konieczny, J. Lang, and P. Marquis. Distance-based merging : a general framework and some complexity results. In *KR*, pages 97–108, 2002.
- [10] S. Konieczny and R. P. Pérez. Merging with integrity constraints. In *ECSQARU*, pages 233–244, 1999.
- [11] G. Ligozat. On generalized interval calculi. In *AAAI*, pages 234–240, 1991.
- [12] G. Ligozat. Reasoning about cardinal directions. *Journal of Visual Languages and Computing*, 9(1) :23–44, 1998.
- [13] J. Lin. Integration of weighted knowledge bases. *Artificial Intelligence*, 83 :363–378, 1996.
- [14] B. Nebel. Solving hard qualitative temporal reasoning problems : Evaluating the efficiency of using the ORD-Horn class. *Constraints*, 1(3) :175–190, 1997.

- [15] D-A. Randell, Z. Cui, and A. Cohn. A spatial logic based on regions and connection. In *KR*, pages 165–176, 1992.
- [16] P. Z. Revesz. On the semantics of arbitration. *Journal of Algebra and Computation*, 7(2) :133–160, 1997.

Annexe : preuves de quelques propositions

Proposition 1. *Si \otimes est une fonction d'agrégation vérifiant le postulat de minimalité forte, alors $\vartheta^{d_r, \otimes}$ est un opérateur de fusion de contraintes, au sens de la définition 1.*

Preuve. Soit \otimes une fonction d'agrégation vérifiant le postulat de minimalité forte. La propriété (a) est vérifiée par définition de $\vartheta^{d_r, \otimes}$, et la propriété (b) est vérifiée puisque \mathcal{B} est un ensemble fini. Nous prouvons la propriété (c) en deux étapes :

- Montrons d'abord $\mu \cap \bigcap_{R \in \mathcal{R}} R \subseteq \vartheta^{d_r, \otimes}(\mu, \mathcal{R})$. Si $\mu \cap \bigcap_{R \in \mathcal{R}} R = \emptyset$, le résultat est trivial. Soit $r \in \mu \cap \bigcap_{R \in \mathcal{R}} R$. $\forall R \in \mathcal{R}$, on obtient $d(r, R) = 0$. Donc $d_{\otimes}(r, \mathcal{R}) = 0$ (minimalité de \otimes). Donc $\forall r' \in \mu$, $d_{\otimes}(r, \mathcal{R}) \leq d_{\otimes}(r', \mathcal{R})$. On a alors $r \in \vartheta^{d_r, \otimes}(\mu, \mathcal{R})$.
- Montrons maintenant $\vartheta^{d_r, \otimes}(\mu, \mathcal{R}) \subseteq \mu \cap \bigcap_{R \in \mathcal{R}} R$. Soit $r \in \vartheta^{d_r, \otimes}(\mu, \mathcal{R})$. On a $r \in \mu$. Supposons $r \notin \bigcap_{R \in \mathcal{R}} R$. $\exists R_h \in \mathcal{R}$ tel que $r \notin R_h$, et donc $d(r, R_h) > 0$. On obtient $d_{\otimes}(r, \mathcal{R}) > 0$ puisque \otimes satisfait le postulat de minimalité forte. Puisque $\mu \cap \bigcap_{R \in \mathcal{R}} R \neq \emptyset$, soit $r' \in \mu \cap \bigcap_{R \in \mathcal{R}} R$, on a $d_{\otimes}(r', \mathcal{R}) = 0$. Donc $\exists r' \in \mu$ $d_{\otimes}(r', \mathcal{R}) < d_{\otimes}(r, \mathcal{R})$. Contradiction avec $r \in \vartheta^{d_r, \otimes}(\mu, \mathcal{R})$. \square

Proposition 2. *N_{res} est cohérent.*

Preuve. Notons N_{res}^k le RCQ construit N_{res} avant l'assignation de la k^e contrainte en respectant l'ordre \prec ; notons alors \mathcal{C}^k , respectivement \mathcal{C}^k l'ensemble des contraintes à fusionner correspondant, respectivement la contrainte C_{ij} à affecter correspondante, notons $N[(k, r)]$ le RCQ $N = (V, C)$ dans lequel on assigne à la contrainte C^k la relation de base r , et enfin notons plutôt μ_k pour la relation d'intégrité μ_{ij} correspondante. Soit $|V_{<}^2|$ la taille de l'ensemble $V_{<}^2$, qui représente également le nombre d'assignations à effectuer au sein de l'algorithme. Montrons par récurrence que $N_{res}^{|V_{<}^2|}$ est cohérent. N_{res}^1 est cohérent par définition. Soit $k > 1$, et supposons N_{res}^k cohérent. On a alors $\exists r \in \mathcal{B} : N_{res}^k[(k, r)]$ est cohérent. Donc $\mu_k \neq \emptyset$. Donc d'après la propriété (b) de la définition 1 d'un opérateur de fusion de contraintes, on a $\vartheta(\mu_k, \mathcal{C}^k) \neq \emptyset$. Par définition de μ_k dans l'algorithme 2 (ligne 10), on a $\forall r \in \mu_k$, $N_{res}^k[(k, r)]$ est cohérent. Or d'après la propriété (a) de la définition 1, $\vartheta(\mu_k, \mathcal{C}^k) \subseteq \mu_k$. Donc $\forall r \in \vartheta(\mu_k, \mathcal{C}^k)$,

$N_{res}^k[(k, r)]$ est cohérent. Ce dernier RCQ correspond au RCQ N_{res}^{k+1} . Par conséquent, par récursivité N_{res} est cohérent. \square

Proposition 3.

$$\bigcap_{N_i \in \mathcal{N}} [N_i] \neq \emptyset \Rightarrow [N_{res}] = \bigcap_{N_i \in \mathcal{N}} [N_i].$$

Preuve. Utilisons les notations de la preuve de la proposition précédente : N_{res}^k est le RCQ construit N_{res} avant l'assignation de la k^e contrainte en respectant l'ordre \prec ; \mathcal{C}^k , respectivement C^k est l'ensemble des contraintes à fusionner correspondant, respectivement la contrainte C_{ij} à affecter correspondante, $N[(k, r)]$ est le RCQ $N = (V, C)$ dans lequel on assigne à la contrainte C^k la relation de base r , et μ_k est la relation d'intégrité μ_{ij} . Comme $\bigcap_{N_i \in \mathcal{N}} [N_i] \neq \emptyset$, il existe un scénario cohérent σ appartenant à $\bigcap_{N_i \in \mathcal{N}} [N_i]$. Soit σ_{ij} la relation de base contenue dans la contrainte du scénario σ entre v_i et v_j , et σ_k la relation de base de σ dans la k^e contrainte à affecter. Montrons par récurrence sur k que pour tout k , $\sigma_k \in \mu_k \cap \bigcap_{C' \in \mathcal{C}^k} C'$. Montrons par récurrence sur k que pour tout k , $\sigma_k \in \mu_k \cap \bigcap_{C' \in \mathcal{C}^k} C'$.

- cas de base : on a déjà $\sigma_1 \in \mu_1 \cap \bigcap_{C' \in \mathcal{C}^1} C'$. En effet puisque $\mu_1 = \mathcal{B}$, $\sigma_1 \in \mu_1$. Comme par définition de σ , $\sigma_1 \in \bigcap_{C' \in \mathcal{C}^1} C'$, on obtient bien $\sigma_1 \in \mu_1 \cap \bigcap_{C' \in \mathcal{C}^1} C'$.
- supposons que pour tout $l \in \{1, \dots, k\}$, on ait $\sigma_l \in \mu_l \cap \bigcap_{C' \in \mathcal{C}^l} C'$. Montrons que $\sigma_{k+1} \in \mu_{k+1} \cap \bigcap_{C' \in \mathcal{C}^{k+1}} C'$. D'après l'hypothèse de récurrence, σ est un sous-réseau de $N^k[(k, \sigma_{k+1})]$, et donc $N^k[(k, \sigma_{k+1})]$ est cohérent. De la définition de μ_{k+1} on obtient $\sigma_{k+1} \in \mu_{k+1}$. Comme par définition de σ , $\sigma_{k+1} \in \bigcap_{C' \in \mathcal{C}^{k+1}} C'$, on obtient $\sigma_{k+1} \in \mu_{k+1} \cap \bigcap_{C' \in \mathcal{C}^{k+1}} C'$.

Il en résulte que pour tout k , $\mu_k \cap \bigcap_{C' \in \mathcal{C}^k} C' \neq \emptyset$. D'après la propriété (c) de la définition 1 d'un opérateur de fusion de contraintes, on a alors pour tout scénario σ et pour tout k , $\sigma_k \in \vartheta(\mu_k, \mathcal{C}^k)$ ssi $\sigma_k \in \mu_k \cap \bigcap_{C' \in \mathcal{C}^k} C' \neq \emptyset$. Ce qui conclut la preuve. \square

Proposition 4. Soit $N = (V, C)$ un RCQ cohérent sur \mathcal{B} et (v_i, v_j) une paire de variables de V . S'il n'existe pas de chaîne de contraintes actives de N C_{ik}, \dots, C_{pj} , alors $\forall r \in \mathcal{B}$, $N[(i, j, r)]$ est cohérent.

Preuve. Soit $N = (V, C)$ un RCQ cohérent sur $2^{\mathcal{B}}$ et (v_i, v_j) une paire de variables de V . Supposons qu'il n'existe pas de chaîne de contraintes actives de N C_{ik}, \dots, C_{pj} . Cela signifie que l'on peut considérer N comme deux RCQ indépendants $N_1 = (V_1, C_1)$ et $N_2 = (V_2, C_2)$ avec $v_i \in V_1$ et $v_j \in V_2$. D'après la supposition faite précédemment, C_{ij} peut être assigné à la relation $\{r\}$ pour tout $r \in \mathcal{B}$, tout en conservant la cohérence de N . \square

Proposition 5. Soit \mathcal{N} un ensemble de RCQ définis sur V , ϑ un opérateur de fusion de contraintes, et \prec un ordre quelconque sur $V_{<}^2$. Si \mathcal{N} est un arbre-ensemble, alors pour toute paire (v_i, v_j) de V contrainte par rapport à \mathcal{N} , $\forall r \in \mathcal{B}$, $N_{res}[(i, j, r)]$ est cohérent.

Preuve. Soit \mathcal{N} un arbre-ensemble de RCQ sur V et (v_i, v_j) une paire de variables contrainte par rapport à \mathcal{N} . Il suffit de montrer qu'il n'existe pas de chaîne de contraintes actives de N_{res} C_{ik}, \dots, C_{pj} , et la preuve découle alors de la proposition 4.

D'après la définition d'un arbre-ensemble, puisque (v_i, v_j) est une paire de variables contrainte par rapport à \mathcal{N} , toute chaîne $C_l = (v_i, v_{k^l}), \dots, (v_{p^l}, v_j)$ contient une paire de variables (v_{a^l}, v_{b^l}) non contrainte par rapport à \mathcal{N} . Dans le graphe $\mathcal{G} = (V, A)$ où l'ensemble des arêtes A correspond à l'ensemble des paires de variables contraintes par rapport à \mathcal{N} , v_i et v_j appartiennent à deux composantes connexes différentes. Donc pour toute chaîne $C_l = (v_i, v_{k^l}), \dots, (v_{p^l}, v_j)$, $\forall r \in \mathcal{B}$, $N_{res}[(a^l, b^l, r)]$ est cohérent. Donc $\mu_{a^l b^l} = \mathcal{B}$. D'autre part toute contrainte de $\mathcal{C}_{a^l b^l}$ est définie par l'ensemble \mathcal{B} . D'après la propriété (c) de la définition 1, $\vartheta(\mu_{a^l b^l}, \mathcal{C}_{a^l b^l}) = \mathcal{B}$. Donc $\mathcal{C}_{a^l b^l} = \mathcal{B}$ est une contrainte inactive. \square

Proposition 6. $\forall (v_i, v_j) \in V_{<}^2$, R_{ij}^* existe et est unique.

Preuve. Soit $(v_i, v_j) \in V_{<}^2$. R_{ij}^* existe, puisque $2^{\mathcal{B}}$ est un ensemble fini et que $\mathcal{B} \in \mathcal{S}$. Prouvons l'unicité de R_{ij}^* . Si $\vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij}) \in \mathcal{S}$, alors $R_{ij}^* = \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$ et la proposition est vérifiée. Sinon, soit R_{ij}^1, R_{ij}^2 deux relations différentes de \mathcal{S} qui s'accordent avec la définition de R_{ij}^* . Soit $R_{ij}^\cap = R_{ij}^1 \cap R_{ij}^2$. R_{ij}^\cap est dans \mathcal{S} , puisque \mathcal{S} est fermé pour l'intersection. De plus, $R_{ij}^\cap \supseteq \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$. Enfin pour $k \in \{1, 2\}$, $d_R(R_{ij}^\cap, \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})) < d_R(R_{ij}^k, \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij}))$, puisque nous supposons que d_R vérifie le postulat d'inégalité stricte (cf. Définition 8). Contradiction. \square

Proposition 8. Si toutes les contraintes des RCQ de \mathcal{N} sont définies par des relations de \mathcal{S} , alors

$$\bigcap_{N_i \in \mathcal{N}} [N_i] \neq \emptyset \Rightarrow [N_{res}^*] = \bigcap_{N_i \in \mathcal{N}} [N_i].$$

Preuve. Puisque \mathcal{S} est fermé pour l'intersection, nous avons $\forall (v_i, v_j) \in V_{<}^2$, $R_{ij}^* = \vartheta^{d_r, \otimes}(\mu_{ij}, \mathcal{C}_{ij})$. La preuve est alors similaire à celle de la proposition 3. \square